Remarks

Claims 1, 3-8, and 10-21 remain in this application. Claims 2 and 9 were previously canceled without prejudice. Claims 1, 8, 15, 16, and 17 are currently amended. No new matter has been added.

Claim Rejections--35 USC 112

Claims 16-17 were rejected as failing to comply with the written description requirement. Applicants respectfully submit that the amended claims now overcomes this rejection.

Regarding claim 16, the term "light-weight" is removed. Regarding claim 17, the terms "a call savings", "an optimization savings", and "a normalized level" are removed.

Therefore, applicants respectfully submit that this rejection is now overcome.

Claim Rejections--35 USC 103

Claims 1, 3-8, and 10-21 stand rejected under 35 U.S.C. § 103 as being unpatentable over Ayers in view of Schmidt. Applicants respectfully submit that the amended claims now overcome this rejections.

Claim 1, as currently amended, recites as follows.

1. A method of compiling a computer program, the method comprising:
   receiving a plurality of modules of source code;
   generating intermediate representations corresponding to the modules;
   extracting a set of data from the intermediate representations to create an inliner summary for each module;
   using the inliner summaries and a globally-sorted working-list based order in an inline analysis phase, without using the intermediate representations in the inline analysis phase, to determine which call sites in the modules are to be inlined by substituting code from a called module, wherein **said globally-sorted**

> working-list based order is <u>dynamically updated</u> during the inline analysis phase; and
> after a call site is determined to be inlined, updating a call graph of the routines and call sites, and updating the inliner summaries throughout the call graph,
> wherein determining the call sites to be inlined involves **proceeding only once through the call sites in said dynamically-updated globally-sorted working-list based order.**

(Emphasis added.)

I. NEITHER AYERS NOR SCHMIDT TEACH **PROCEEDING ONLY ONCE** THROUGH THE WORKING LIST OF CALL SITES

Claim 1 recites that "determining the call sites to be inlined involves **proceeding only once** through the call sites in said dynamically-updated globally-sorted working-list based order." (Emphasis added) This claim limitation is supported, for example, in the original application by the flow chart of FIG. 5 and the description thereof on pages 19-20.

In contrast, Ayers teaches a multi-pass inliner which performs an inline analysis multiple times. See, for example, Ayers, page 135, right column, line 5, "HLO performs **several passes** of inlining and cloning." (Emphasis added.)

Schmidt proceeds through the call sites in **multiple rounds**. The **initial round** uses the initial priority queue. (See col. 5, line 59, "In the first approximation phase ....") In the **second round**, the technique traverses from the leaves towards the root of the call-graph and compares the profit of the already-accepted call-sites with the profit from the elements in AuxQueue. The elements in AuxQueue are the not-already-inlined call sites that were encountered during this upward traversal. Depending on the comparison, the second round of Schmidt refines the decisions made in the initial round. (See col. 6, line 1, "In the refinement stage ....") Thus, Schmidt clearly proceeds through the call sites in multiple rounds, while claim 1 recites "only once". .

Therefore, for at least this reason, applicants respectfully submit that amended claim 1 is patentably distinguished over Ayers in view of Schmidt.

II.   NEITHER AYERS NOR SCHMIDT TEACH **DYNAMICALLY UPDATING** THE WORKING LIST OF CALL SITES

Applicants respectfully submit that the claim limitation that "said globally-sorted working-list based order is **dynamically updated** during the inline analysis phase" (emphasis added) is contrary to the disclosure and teachings of both Ayers and Schmidt.

This claim limitation is supported, for example, in the original application at page 23, lines 29-32, which is reproduced below for convenience of reference.

> If the caller or the callee routine for a certain call site has its summary information modified, the goodness factor of that call site is recomputed. **Whenever the goodness factor of a certain call site is recomputed, the old entry is deleted from the working list and a new updated entry is inserted.**
> (Emphasis added.)

In other words, the working list for the inline analysis phase is dynamically updated based on changes in the goodness factor of call sites.

In contrast, Ayers teaches using an initial (statically sorted) list throughout to perform the inline analysis.   This is supported by the following citation. Ayers, section 2.4, third paragraph, lines 1-4, states as follows. "**The inliner then walks over the inline site list in priority order.** The compile-time impact of each site is considered, and if within the current budget, the inline is accepted." (Emphasis added.) Therefore, Ayers teaches going through the inline site list in a static "priority" order. This is *contrary* to this claim limitation reciting a **dynamically updated** working-list based order.

Similarly, Schmidt teaches using an initial priority queue which is statically calculated to go through the call sites (called "arcs" in Schmidt) and then discarding that initial priority queue. This is clearly stated in column 5, lines 44-65, which recites as follows. "**In the first approximation phase of FIG. 2, all inline candidates or arcs with finite priority are placed in an initial priority queue.** Then as indicated at a block 306, the best inline candidates are removed from the initial priority queue, marking them as tentatively to be inlined, until the code bloat

budget has been exhausted. **This initial queue is then discarded.**" (Emphasis added.) Therefore, Schmidt teaches going through the call sites according to an "initial priority queue". There appears <u>no</u> disclosure in Schmidt as to the claimed **dynamically updated** working-list based order.

Hence, for at least this further reason, applicants respectfully submit that amended claim 1 is patentably distinguished over Ayers in view of Schmidt.

Claims 3-7 depend from claim 1. Therefore, applicants respectfully submit that claims 3-7 overcome their rejections for at least the same reasons as discussed above in relation to claim 1.

Claim 8 recites both the "dynamically-updated" limitation and "proceeding only once" limitation. Therefore, applicants respectfully submit that claim 8 overcomes its rejection.

Claims 10-14 depend from claim 8. Therefore, applicants respectfully submit that claims 10-14 overcome their rejections for at least the same reasons as discussed above in relation to claim 8.

Claim 15 recites both the "dynamically-updated" limitation and "proceeding only once" limitation. Therefore, applicants respectfully submit that claim 8 overcomes its rejection.

Claim 16 recites both the "dynamically updating" limitation and "a one-pass inline analysis phase" limitation. Therefore, applicants respectfully submit that claim 16 overcomes its rejection.

Claims 17-21 depend from claim 16. Therefore, applicants respectfully submit that claims 17-21 overcome their rejections for at least the same reasons as discussed above in relation to claim 16.
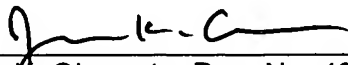
Conclusion

For the above-discussed reasons, applicant respectfully submits that the pending claims, as hereby amended, now overcome the rejections of the latest office action and are now in form for allowance. Favorable action is respectfully requested.
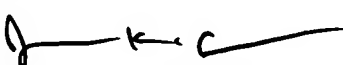
The Examiner is also invited to call the below-referenced attorney to discuss this case.

Respectfully Submitted,

Dhruva R. Chakrabarti et al.

Dated: March 10, 2008   By: _____

James K. Okamoto, Reg. No. 40,110
Tel: (408) 436-2111
Fax: (408) 436-2114